

CLAIMS

What is claimed is:

- 1 1. A method comprising:
 - 2 receiving a binary of a program code, the binary based on a first instruction set
 - 3 architecture; and
 - 4 translating the binary, wherein the translated binary is based on a combination
 - 5 of the first instruction set architecture and a second instruction set architecture.

- 1 2. The method of claim 1, comprising checking instruction set architecture execution flags, the instruction set architecture execution flags to indicate at least one translation of a portion of the binary.

- 1 3. The method of claim 2, wherein the instruction set architecture execution flags
2 are set by a programming environment of the binary.

- 1 4. The method of claim 2, wherein a register in a processor translating the binary
2 is to store the instruction set architecture execution flags.

- 1 5. The method of claim 2, comprising executing the translated binary.

- 1 6. The method of claim 5, wherein the translating and executing are based on a
2 command, the instruction set architecture execution flags based on a number of
3 command line flags associated with the command.

- 1 7. The method of claim 1, wherein the first instruction set architecture comprises
2 floating-point instructions and wherein the second instruction set architecture
3 comprises floating-point instructions, wherein the translating of the binary comprises

4 translating the floating-point instructions of the first instruction set architecture to the
5 floating-point instructions of the second instruction set architecture.

1 8. The method of claim 1, wherein the translating of the binary comprises storing
2 a portion of a hardware stack in a register of a processor translating the binary.

1 9. A method comprising:
2 receiving a binary of a program code, the binary based on a first instruction set
3 architecture; and
4 executing the binary, wherein the executing comprises translating at least one
5 instruction of the binary based on the first instruction set architecture to at least one
6 instruction based on a second instruction set architecture.

1 10. The method of claim 9, wherein the first instruction set architecture includes
2 in-order accesses to memory and the second instruction set architecture includes out-
3 of-order accesses to memory, the translating of the binary to include out-of-order
4 accesses to memory by a processor executing the binary.

1 11. The method of claim 9, wherein the first instruction set architecture allows the
2 binary to self modify and the second instruction set architecture does not allow the
3 binary to self modify, the translating of the binary to include an instruction to
4 controllers of memories that store the binary to perform write operations independent
5 of checks of whether the write operations modify a location where the binary is
6 stored.

1 12. The method of claim 9, wherein the second instruction set architecture has an
2 address space that is larger than the first instruction set architecture, the translating of
3 the binary comprises using the address space of the second instruction set
4 architecture.

10039255.092020

1 13. The method of claim 12, wherein data accessed by the binary is stored in a
2 single segment in memory and wherein an offset value for translating a virtual address
3 to a physical address for the data is not modified during execution of the binary.

1 14. A system comprising:
2 a memory to include a binary of a program code based on a first instruction set
3 architecture; and
4 a processor coupled to the memory, the processor to execute the binary,
5 wherein executing the binary comprises translating the binary, the translated binary
6 based on a combination of the first instruction set architecture and a second
7 instruction set architecture.

1 15. The system of claim 14, wherein the processor comprises a register to store
2 instruction set architecture execution flags, the instruction set architecture execution
3 flags to indicate at least one translation of a portion of the binary.

1 16. The system of claim 15, wherein the instruction set architecture execution
2 flags are set by a programming environment of the binary.

1 17. The system of claim 14, wherein the second instruction set architecture has an
2 address space that is larger than the first instruction set architecture, the translating of
3 the binary comprises using the address space of the second instruction set
4 architecture.

1 18. The system of claim 17, wherein the binary is stored in a single segment in the
2 memory and wherein an offset value for translating a virtual address to a physical
3 address is not modified during execution of the binary.

1000000000000000

1 19. An apparatus comprising:
2 a decoder to receive a binary based on a first instruction set architecture; and
3 a number of registers, wherein at least one of the number of registers is to
4 store instruction set architecture execution flags, the instruction set architecture
5 execution flags to indicate a translation of a binary, the translated binary based on a
6 combination of the first instruction set architecture and a second instruction set
7 architecture.

1 20. The apparatus of claim 19, wherein the first instruction set architecture
2 comprises floating-point instructions and wherein the second instruction set
3 architecture comprises floating-point instructions, wherein the translating of the
4 binary comprises translating the floating-point instructions of the first instruction set
5 architecture to the floating-point instructions of the second instruction set architecture.

1 21. The apparatus of claim 19, wherein the translating of the binary comprises
2 storing a portion of a hardware stack in a register within the number of registers.

1 22. The apparatus of claim 19, wherein the apparatus is coupled to memories to
2 store the binary, wherein the first instruction set architecture allows the binary to self
3 modify and the second instruction set architecture does not allow the binary to self
4 modify, the translating of the binary to include an instruction to controllers of the
5 memories to perform write operations independent of checks of whether the write
6 operations modify a location where the binary is stored.

1 23. The apparatus of claim 19, wherein the second instruction set architecture has
2 an address space that is larger than the first instruction set architecture, the translating
3 of the binary comprises using the address space of the second instruction set
4 architecture.

1 24. The apparatus of claim 23, wherein data accessed by the binary is stored in a
2 single segment in memory coupled to the apparatus and wherein an offset value for
3 translating a virtual address to a physical address for the data is not modified during
4 execution of the binary.

1 25. A machine-readable medium that provides instructions, which when executed
2 by a machine, causes the machine to perform operations comprising:
3 receiving a binary of a program code, the binary based on a first instruction set
4 architecture; and

5 translating the binary, wherein the translated binary is based on a combination
6 of the first instruction set architecture and a second instruction set architecture.

1 26. The machine-readable medium of claim 25, comprising executing the
2 translated binary.

1 27. The machine-readable medium of claim 26, wherein the translating and
2 executing are based on a command, the instruction set architecture execution flags
3 based on a number of command line flags associated with the command.

1 28. The machine-readable medium of claim 25, wherein the first instruction set
2 architecture comprises floating-point instructions and wherein the second instruction
3 set architecture comprises floating-point instructions, wherein the translating of the
4 binary comprises translating the floating-point instructions of the first instruction set
5 architecture to the floating-point instructions of the second instruction set architecture.

1 29. The machine-readable medium of claim 25, wherein the first instruction set
2 architecture allows the binary to self modify and the second instruction set
3 architecture does not allow the binary to self modify, the translating of the binary to
4 include an instruction to controllers of memories that store the binary to perform write
5 operations independent of checks of whether the write operations modify a location
6 where the binary is stored.